1. 数据库操作

创建数据库:

```
create database mydb;
create databse if not exists mydb comment 'create my db named dbname' with
dbproperties ('a'='aaa','b'='bbb')
```

查看创建语句:

```
show create database mydb;
```

查询数据库的详细信息:

```
desc database extended mydb;
```

列出所有的数据库:

```
show databases;
```

删除数据库:

```
drop database mydb;
drop database if exists mydb cascade;
```

使用数据库:

```
use mydb;
```

查看数据库的数据表:

```
show tables;
show tables in mydb;
```

2. 添加表

创建内部表（Managered_Table）

```
create table mingxing(id int, name string, sex string, age int, department
string) row format delimited fields terminated by ',';
```

创建外部表（External_Table）

```
create external table mingxing(id int, name string, sex string, age int,
department string) row format delimited fields terminated by ',' location
'/root/hivedata';
```

注意：创建外部表的时候指定location的位置必须是目录，不能是单个文件

创建分区表（Partition_Table）

```
create table mingxing(id int, name string, sex string, age int, department
string) partitioned by (city string) row format delimited fields terminated by
',';
```

注意：partition里的字段不是能是表中声明的字段

创建分桶表（Bucket_Table）

```
create table mingxing(id int, name string, sex string, age int, department
string) clustered by(id) sorted by(age desc) into 4 buckets row format delimited
fields terminated by ',';
```

3. 删除表

删除表：

```
drop table [if exists] mingxing;
```

4. 对表进行重命名

重命名数据库表：

```
alter table mingxing rename to student;
```

5. 修改表

增加字段：

```
alter table mingxing add columns (province string);
alter table mingxing add columns (province string, xxx bigint);
```

删除字段：

```
drop（不支持），替代方案可以使用replace
```

修改字段：

```
alter table mingxing change age newage string;
alter table mingxing change age newage string first|after id;
```

替换字段

```
alter table mingxing replace columns(id int, name string, sex string);
```

6. hive分区表

增加分区：

```
alter table mingxing add partition(city='beijing');
alter table mingxing add partition(city='beijing') partition(city='tianjin');
```

删除分区：

```
alter table mingxing drop if exists partition(city='beijing');
```

修改分区数据路径：

```
alter table mingxing partition(city='beijing') set location
'/home/hadoop/data/beijing';
```

7. 各种常用查询显示命令

查看库：

```
show databases;
```

查看建库语句：

```
show create database mydb;
```

查看表：

```
show tables;
show tables in mydb;
```

查看建表语句：

```
show create table mingxing;
```

查看内置函数库：

```
show functions;
```

查看分区：

```
show partitions mingxing;
```

查看表的字段：

```
desc mingxing;
```

查看表的详细信息：

```
desc extended mingxing;
```

查看表的格式化了之后的详细信息：

```
desc formatted mingxing;
```

8. load方式导入数据

导入本地相对路径的数据

```
load data local inpath './student.txt' into table mingxing;
load data local inpath './student.txt' overwrite into table mingxing;       # 覆盖
导入
```

导入本地绝对路径数据：

```
load data local inpath '/root/hivedata/student.txt' into table mingxing;
```

导入HDFS上的简便路径数据：**

```
load data inpath '/root/hivedata/student.txt' into table mingxing;
```

导入HDFS上的全路径模式下的数据：**

```
load data inpath 'hdfs://hadoop01:9000/root/hivedata/student.txt' into table
mingxing;
```

9. 利用insert关键字往表中插入数据

单条数据插入：

```
insert into table mingxing values(001,'huangbo','male',50,'MA');
```

单重插入模式：

```
insert into table student select id,name,sex,age,department from mingxing;
```

注意：查询出的字段必须是student表中存在的字段

多重插入模式：

```
from mingxing
insert into table student1 select id,name,sex,age
insert into table student2 select id,department;
```

```
from mingxing2
insert into table student1 partition(department='MA') select id,name,sex ,age
where department='MA'
insert into table student1 partition(department='CS') select id,name,sex ,age
where department='CS';
```

静态分区插入：

```
load data local inpath '/root/hivedata/student.txt' into table student
partition(city='henan');
```

动态分区插入：

```
create table student(name string, department string) partitioned by (id int)
.....
insert into table student partition(id) select name,department,id from
mingxing2;
```

student表字段：name,department， 分区字段是id

查询字段是：name,department,id，分区字段

注意：动态分区插入的分区字段必须是查询语句当中出现的字段中的最后一个

CTAS(create table ... as select ...)

直接把查询出来的结果存储到新建的一张表里

```
create table student as select id,name,age,department from mingxing;
```

注意：自动新建的表中的字段和查询语句出现的字段的名称，类型，注释一模一样

10. like关键字使用

仿造表：

```
create table student like mingxing;
```

11. 利用insert导出数据到本地或者hdfs

单模式导出数据到本地：

```
insert overwrite local directory '/root/outputdata' select
id,name,sex,age,department from mingxing;
```

多模式导出数据到本地：

```
from mingxing
insert overwrite local directory '/root/outputdata1' select id, name
insert overwrite local directory '/root/outputdata2' select id, name,age
```

简便路径模式导出到hdfs**：**

```
insert overwrite directory '/root/outputdata' select id,name,sex,age,department
from mingxing;
```

全路径模式查询数据到hdfs：**

```
insert overwrite directory 'hdfs://hadoop01:9000/root/outputdata1' select
id,name,sex,age,department from mingxing;
```

12. 清空数据库表

清空表中的数据：

```
truncate table mingxing;
```

13. 查询数据

基本查询：

```
select * from mingxing join student where ... group by ... order by ... limit
...
```

查询全局有序数据：

```
select * from mingxing order by age desc , id asc;
```

如果数据量过大，我们**采用局部排序**的方式：

```
set mapred.reduce.tasks=3;
set mapreduce.job.reduces=3;
select * from mingxing sort by id asc;
```

分桶查询：

```
set hive.enforce.bucketing = true;
select * from mingxing distribute by sex;
```

查询排序的分桶数据：

```
select * from mingxing cluster by id sort by id desc, age asc;
```

14. 五种链接查询

内连接inner join：

```
select student.*, mingxing.* from student join mingxing on student.id =
mingxing.id
select student.*, mingxing.* from student inner join mingxing on student.id =
mingxing.id
```

左外链接left outer join:

```
select student.*, mingxing.* from student left join mingxing on student.id =
mingxing.id
select student.*, mingxing.* from student left outer join mingxing on student.id
= mingxing.id
```

右外链接right outer join:

```
select student.*, mingxing.* from student right join mingxing on student.id =
mingxing.id
select student.*, mingxing.* from student right outer join mingxing on
student.id = mingxing.id
```

全外链接full outer join:

```
select student.*, mingxing.* from student full join mingxing on student.id =
mingxing.id
select student.*, mingxing.* from student full outer join mingxing on student.id
= mingxing.id
```

in/exists的hive高效实现left semi join：

```
select student.*, mingxing.* from student left semi join mingxing on student.id
= mingxing.id;
```

等同于：

```
select student.* from student where student.id in(select distinct id from
mingxing);
```